UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/618,810	07/14/2003	Ajay Kumar	5681-15200	5782

58467 7590 04/28/2009

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
WONG, JOSEPH D	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
04/28/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/618,810	KUMAR ET AL.	
	Examiner	Art Unit	
	JOSEPH D. WONG	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 February 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-11 and 13-58* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-11 and 13-58* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 Feb 2009* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1, 3-11, 13-56 are amended.

Claims 2 and 12 are canceled.

Claims 57-58 are new.

Response to Arguments

Double Patenting

On page 19 of the instant remarks, Applicant's arguments countering a provisional rejection of obvious-type double patenting and instant claims are reconsidered but are deemed not persuasive. Since any terminal disclaimer requires additional review beyond that of the Examiner, Applicant is advised that any delay from Applicant in submitting an approvable terminal disclaimer or distinguishing the claims with nonobvious variations may contribute to significant delays which would be unfortunate if the case were close to a particular deadline requiring a step to be taken in a later stage of prosecution. After review, this ground of rejection is maintained.

Claim objections

Claim objections under 37 CFR 1.83 are withdrawn in view of an instant replacement drawing.

Drawings

On page 19, paragraphs 1-3, Applicant argued that under 37 CFR 1.84(q) the objected drawings are really a surface or cross section. However, mere citation of 37 CFR 1.84(q) is not instantly persuasive as the reference numerals in dispute are in a block diagram or flow chart without perspective and not necessarily shown upon a surface or cross sectional view. Note Fig. 10, reference numeral 1015 has a formal curved lead line which complies with 37 CFR 1.84(q). Therefore, instant objections under 37 CFR 1.84(p)(1) to the drawings instantly stand.

Specification

Objection to the abstract is withdrawn in view of an amended abstract submitted.

Rejection under 35 U.S.C. §101

Rejections under 35 USC 101 are withdrawn.

Rejection under 35 U.S.C. §102

On page 20, last paragraph through page 27, third paragraph, Applicant argues that the prior art does not teach "one or more transaction managers configured to control state changes of the one or more atomic transactions initiated by the one or more applications, wherein for each

given atomic transaction, the one or more transaction managers are configured to request a read lock on a transaction freeze object to change the state the given atomic transaction". Applicant further argues that the cited art does not request a read lock on a transaction freeze object to change the state of the given atomic transaction. Applicant additionally argues that the prior art does not teach "transaction freeze object" and "read locks on transaction freeze objects". However, these arguments are moot in view of a new ground of rejection under 35 USC 103(a) adding Fowler to Hagersten. Fowler teaches in Col. 5, Lines 40-53,"pause/resume synchronization" where a pause necessarily implies a transaction freeze. Furthermore, Fowler teaches in Col. 8, Lines 52-56, "control signal to...effect resumption of program execution". Additionally Fowler's resume feature in Col. 5, Lines 40-53 of pause/resume synchronization necessarily implies permission to change the state of a transaction freeze by obtaining first a read lock and then a write lock on a transaction freeze object to effect a change in state. Therefore claims 1, 10, 11 and 20 stand rejected.

On pages 27 through 29, Applicant argues that the prior art does not teach "receiving a pause request, and pausing a transaction manager in response to the pause request by withholding permission to change the state of one or more transactions managed by the transaction manager". Applicant further argues that Oeltjen has nothing to do with pausing a transaction manager, much less pausing a transaction manager by withholding permission to change the state of one or more transaction managed by the transaction manager. Applicant further argues that Oeltjen does not teach "receiving a plurality of resume requests, and resuming the transaction manager in response to the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager". However, Oeltjen

expressly recites debug commands "pause, stop or kill a task" in paragraph [38]. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "...receiving a pause request, and pausing a transaction manager in response to the pause request by withholding permission to change the state of one or more transactions managed by the transaction manager") is not commensurate with the scope of what is recited in the rejected claim 21. Claim 21 recites the argued limitation behind a preface of "configured to" which means that the claim does not necessarily do the step but needs to be capable of doing the steps. In arguendo, Oeltjen, paragraph [9], fixing one tool and causing another tool to fail testing is a coincidental way to withhold permission to change the state of one or more transactions as the claim does not require that the withholding of the permission is necessarily and only responsive to the pause request. Pausing is responsive to the pause command but there is nothing in the claim that requires the withholding of the permission to be exclusively bound to the pause command. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Oeltjen recites in paragraph [38], a plurality of resume requests such as "step, next, run, continue". Therefore claims 21, and 39 stand rejected.

Rejection under U.S.C. §103

On pages 30 through 31, Applicant argues that the prior art does not teach "pausing a transaction manager in response to the pause request **by** withholding read locks on a stored transaction freeze object that identifies a respective atomic transaction". However, in response

to applicant's argument that the references fail to show certain features of applicant's invention, the preposition "by" can more broadly refer to the vicinity or coincident rather than direct causal relationship. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Therefore claim 30 stands instantly rejected.

On page 31, paragraph 2, Applicant argues that the motivation is not supported by the evidence of record. Applicant further argues that combining the teachings of Oliver with Ault would completely undermine a primary goal of the Ault reference. Applicant goes on to argue that "[n]o one of ordinary skill in the art would combine the teachings of the cited art for the reason presented". However, the motivation cited by the Examiner is a verbatim quote cited from Oliver's Abstract. Applicant's characterization of the Ault reference is incorrect that Ault is necessarily limited to serializing access to a shared resource. Ault serializes access to a semaphore counter but this limited serialization of a semaphore counter is consistent with guarding counters that support multiple reader access and single writer access as evinced by non-binary semaphores or higher value counters. Therefore allegations of a teaching away are not supported by a controlling citation to counter a common subclass classification cross reference to 709/107 for Ault and Oliver. Therefore a person having ordinary skill in the art would have been motivated at the time of invention having Ault and Oliver before him or her in the same shoe to make the combination.

For at least the reasons above, all pending claims stand rejected.

Drawings

Figures 1-10 are objected to because they are not formal with respect to 37 CFR 1.84(p)(1) because the reference numerals are enclosed within an outline. Note Fig. 10, reference numeral 1015 has a formal curved lead line. Mere citation of 37 CFR 1.84(q) is not instantly persuasive as the reference numerals in dispute are in a block diagram or flow chart and not necessarily shown upon a surface or cross sectional view.

Claim Objections

Claims 30 and 48 are objected under 37 CFR 1.83 for not appearing to illustrate the feature "resuming the transaction manager in response the resume request by granting read locks on the transaction freeze object". Consequently dependent claims 31-38, 48-56 are objected also.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting

ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1, 10, 11, 20, 21, 30, 39 and 48 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-36 of copending Application No. 10/618,828. Although the conflicting claims are not identical, they are not patentably distinct from each other because it appears commonly owned with common inventors and the claimed subject matter is appears to differ superficially. In claim 1, the instant application adds, "wherein for each transaction". Other synonymous phraseology or obvious variations are present such as instant claim 1 recites "not change the state of the transaction without said permission" and the copending claim 1 recites "does not allow the one or more transactions to complete".

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Information Disclosure Statement

The references that were cited in copending application 10/618,828 have been partially considered on a best effort basis, but are not necessarily considered as a whole unless Applicant provides any pertinent references cited in prosecution with another Examiner on a separate list in compliance with 37 CFR 1.98(a)(1). In order to have the references printed on any such resulting patent, a separate listing, preferably on a PTO/SB/08A and 08B form, must be filed within the set period for reply to this Office action. If Applicant wishes to have references

commonly cited in the instant application cited and copending application 10/618,828, Applicant can file an IDS.

Claim Objections

Claims 21, 30, 39 and 48 are objected to for having minor punctuation informalities under MPEP 608.01(m).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 21-25, 27, 29 are rejected under 35 U.S.C. 102(e) as being anticipated by Oeltjen et al, (US 2004/0225972 A1), hereinafter Oeltjen.

As to claim 21, Oeltjen teaches a method, comprising: using one or more computers to perform: receiving a pause request ([38], "commands that permit stepping through and editing code such as pause"); pausing a transaction manager in response to the pause request by

withholding permission ([9], "all this is done manually and is understandably prone to many mistakes, both human and resulting from the interactions of the many components of the design...more errors result from tools failing to interact properly...yet...errors require debugging", since a compile or compatibility error necessarily prevent the program from starting , such an error necessarily blocks a pause or restart) to change the state of one or more transactions managed by the transaction manager ([38], "permit stepping...step...pause...kill"), receiving a plurality of resume requests ([38], "commands such as...next, ..., continue"); and resuming the transaction manager in response the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager ([38], "permit stepping through").

As to claim 22, Oeltjen teaches .the method as recited, wherein a transaction freeze manager grants and withholds said permission (Fig. 7, item 718, "result...fail", where a test failure ends execution in a manner that excludes pause and resume options; see Fig. 1, item 52).

As to claim 23, Oeltjen teaches the method as recited, wherein the transaction freeze manager is a part of the transaction manager (Fig. 4, items 440, 470, 474, [38]).

As to claim 24, Oeltjen teaches the method as recited, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager from an administrative entity ([38], "flow manager 460 also gives a flow developer 424 or other user full control over the location within the flow to manually set the current flow").

As to claim 25, Oeltjen teaches the method as recited, wherein the transaction freeze manager is (interpreted to be a capability) configured to queue received state transition

permission requests and transaction manager pause requests in the order received ([38], See Fig. 7, where queuing is broadly interpreted to include any depth of buffering even 1 because all operations within the real world require time and are not instantaneously, see [35]).

As to claim 27, Oeltjen teaches the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a state transition permission request (conditional) if the transaction manager is not paused (Fig. 7, [38], "specifically commands that permit stepping through and editing code such as step, next, run, continue, as well as pause").

As to claim 29, Oeltjen teaches the method as recited, wherein the transaction freeze manager is configured to not grant requests if the transaction manager is paused ([37], "without this feature and given that flows may call hundreds of tools and take weeks to execute, any change or disability of the executing computers would require starting the process again").

Claims 39-41, 43, 45, 47 are rejected under 35 U.S.C. 102(b) as being anticipated over Ault et al, (US 6,237,019), hereinafter Ault.

As to claim 39, Ault teaches a computer readable storage medium storing program instructions, wherein the program instructions are computer-executable to: receive a pause request (Fig. 3); pause a transaction manager in response to the pause request (Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue...by creating a queues") by withholding permission to change the state of one or more transactions managed by the transaction manager; receive a resume request (Fig. 2, "ACCESS RESOURCE SEMOP(SEMID,+1); Fig. 3, "GRANT SEMAPHORE", item 316 is a read lock); and resume the

transaction manager in response to the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager (Fig. 2, SEMOP(SEMID, -1)->"GRANT ACCESS", item 222).

As to claim 40, Ault teaches the computer readable storage medium as recited, wherein a transaction freeze manager grants and withholds said permission (Fig. 3, Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue").

As to claim 41, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is a part of the transaction manager (Col. 2, Lines 48-62).

As to claim 43, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received (Col. 2, Lines 40-43, "in order for the kernel semaphore logic to atomically update the semaphore value and maintain the wait queue").

As to claim 45, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to grant a state transition permission request (conditional) if the transaction manager is not paused (Fig. 2-3, 5A-5E).

As to claim 47, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to not grant requests (conditional) if the transaction manager is paused (Col. 2, Lines 44-47, "...when the kernel owns this lock, other callers of semop(sic) will be suspended waiting for this lock..").

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this claim, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3-11, 13-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hagersten et al, (US 5,983,326), hereinafter Hagersten in view of Fowler et al, (US 4,502,116), hereinafter Fowler.

As to claim 1, Hagersten teaches a system (Fig. 1), comprising: one or more processors (Col. 4, Lines 51-56, "multiprocessing computer system includes a plurality of processing nodes interconnected by an interconnect network"; Col. 7, Lines 14-16, " SMP node 12 includes multiple processors"); memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement: one or more applications configured to initiate one or more atomic transactions (Col. 8, Lines 20-24, "memory operation...causing transfer of data from a source to a destination...within the initiator"; Col. 9, Lines 10-11, "Memory 22 is configured to store data and instruction code for use by processors"), wherein each of the one or more atomic transactions comprises requests to access one or more data sources (Fig. 1, item 22, "Memory", nodes 12A-12D; Col. 9, Lines 6-9, "address and data phases of a transaction may be identified via a unique tag"); and a transaction manager configured to control state changes of the one or more atomic transactions initiated by

the one or more applications (Col. 4, Lines 62-66, "home agent is configured to service multiple requests simultaneously...transaction blocking unit is coupled to a home agent control unit for preventing the servicing of a pending coherent transaction request"); wherein for each given atomic transaction (Col. 1, Lines 19-22; Col. 8, Lines 26-27, read and write operations), the transaction manager is configured to request permission to change the state of the transaction (Fig. 7, "Receive Request...Check Block Status...Set Blocked Status...Write Reply...Write Data...Clear Block Status"), and wherein the transaction manager is configured to not change the state of the given atomic transaction without said permission (Fig. 11, "NACK...Not Acknowledge" or "NOPE...Negative Response").

However, Hagersten strongly suggests but does not expressly teach a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction (Fig.14, 15A, "transaction blocking unit", Col. 3, Lines 10-15, Lines 32-38, "spin lock").

Fowler teaches a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction (Col. 5, Lines 40-53,"pause/resume synchronization of the multiprocessor system"; Col. 8, Lines 52-56, "control signal to...effect resumption of program execution").

Hagersten and Fowler are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Hagersten and Fowler because it provides for a stable testing and debugging environment for a multiprocessor system that aids in the debugging of the total system as discussed in Fowler, Col. 1, Lines 66-67; Col. 2, Lines 1-2.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Hagersten and Fowler because it provides for a stable testing and debugging environment for a multiprocessor system that aids in the debugging of the total system as suggested in Fowler, Col. 1, Lines 66-67; Col. 2, Lines 1-2.

As to claim 3, Hagersten teaches the system as recited, wherein the transaction freeze manager is a part of the transaction manager (Col. 3, Lines 5-15; Col. 30-50).

As to claim 4, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to receive requests to pause the transaction manager from an administrative entity (Fig. 9, 10"ADM...Administrative").

As to claim 5, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received (Fig. 15A, item 406, "RTS"; Col. 11, Lines 47-53, "FIFO").

As to claim 6, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11,Lines 47-53, "FIFO; Col 26, Line 40).

As to claim 7, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to grant the said permission request in response to determining that the transaction manager is not paused (Fig. 7, "not blocked"->"set blocked status").

As to claim 8, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to grant the pause request in response to determining that the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request (Fig. 7, "not blocked"->"set blocked status").

As to claim 9, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to not grant requests (conditional) if the transaction manager is paused (Fig. 7, "not blocked"->"set blocked status").

As to claim 10, Hagersten teaches a system (Fig. 1), comprising a plurality of computer systems coupled by one or more networks (Fig 1B, item 38, "network"; Fig. 1, item 24, "System interface", item 20, "SMP BUS 20" meets network), wherein the plurality of computer systems comprise: one or more processors (Fig. 1, items 16A-16B, "P...P" for processors); and memory (Fig. 1, item 22, "memory") coupled to the one or more processors (Fig. 1, item 16A-16B, "P...P") and configured to store program instructions (Col. 3, Lines 10-15, 33-38, "if failed begin goto top end") executable by the one or more processors to implement one or more application servers comprising: one or more applications (Fig. 3, item 102, "home agent"; item 100, "request agent"; item 98, "transaction filter") configured to initiate one or more atomic transactions (Col. 6, Lines 30-33, "operation performed in response to a read to own request

from a processor"; Fig. 1-2), wherein each of the one or more atomic transactions comprises requests to access one or more data sources (Fig. 1, item 22A-D, "memory" , nodes 12A-12D); one or more transaction managers configured to control state changes of the one or more atomic transactions initiated by the one or more applications (Fig. 3, item 102, "home agent"...item 100, "request agent"); wherein for each atomic transaction (Fig. 4, item 110, "request"; Fig. 6, "request active"), the one or more transaction managers are configured to request permission to change the state of the given atomic transaction (Fig. 6, item 144, "Request Active"; item 150, "Write Active"); and wherein the one or more transaction managers are configured to not change the state of the given atomic transaction without said permission (Fig. 6, "Ignored Write").

However, Hagersten strongly suggests but does not expressly teach a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction (Fig.14, 15A, "transaction blocking unit", Col. 3, Lines 10-15, Lines 32-38, "spin lock").

Fowler as applied above teaches a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction (Col. 5, Lines 40-53,"pause/resume synchronization of the multiprocessor system"; Col. 8, Lines 52-56, "control signal to...effect resumption of program execution").

As to claim 11, Hagersten teaches a system, comprising: one or more processors (Fig. 1); memory (Fig. 1, item 22, "memory") coupled to the one or more processors (Fig. 1, item 16A-B, "P....P") and configured to store program instructions executable by the one or more processors to implement: one or more applications (Fig. 3, "home agent") configured to initiate one or more atomic transactions (Fig. 3, item 102, "home agent" or item 100, "request agent" or item 104, "slave agent"), wherein each of the one or more atomic transactions comprises requests to access one or more data sources (Fig. 1, item 22A-D, "memory"); a transaction manager configured to control state changes of the one or more transactions initiated by the one or more applications (Fig. 3, item 102, "home agent"...item 100, "request agent"); wherein for each transaction ( ), the transaction manager is configured to request a read lock on a stored transaction freeze object to change the state of the given atomic transaction (Fig. 6, item 144, "Request Active"; item 150, "Write Active"); wherein the transaction manager is configured to not change the state of the transaction without said lock (Fig. 6, "Ignored Write").

However, Hagersten strongly suggests but does not expressly teach a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said read lock for said stored transaction freeze object (Fig.14, 15A, "transaction blocking unit", Col. 3, Lines 10-15, Lines 32-38, "spin lock").

Fowler as applied above teaches a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to

resume the transaction manager in response to a resume request by granting said read lock for said stored transaction freeze object (Col. 5, Lines 40-53,"pause/resume synchronization of the multiprocessor system"; Col. 8, Lines 52-56, "control signal to...effect resumption of program execution").

As to claim 13, see claim 3 above.

As to claim 14, see claim 4 above.

As to claim 15, see claim 5 above.

As to claim 16, see claim 6 above.

As to claim 17, see claim 7 above.

As to claim 18, see claim 8 above.

As to claim 19, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to not grant locks in response to determining a write lock on the transaction freeze object is currently held by an administrative entity (Fig. 9-10).

As to claim 20, Hagersten teaches a system (Fig. 1), comprising a plurality of computer systems coupled by one or more networks (Fig. 1A, item 38, "network"; Fig.1, item 20, "bus" or "system interface"), wherein the plurality of computer systems comprise: one or more processors (Fig. 1, items 16A-16B, "P....P", for processors); and memory (Fig. 1, item 22, "memory") coupled (Fig. 1, item 20, "bus") to the one or more processors (Fig. 1, item 16A-B, "P....P") and configured to store program instructions executable (Fig. 9, 10, "read to share" or "read to

own"; where read and own are instructions) by the one or more processors (Fig. 1, items 16A-B) to implement one or more application servers (Fig. 15A, "home agent control unit"; Fig. 15A, "directory cache management unit") comprising: one or more applications (Fig. 3, "home agent") configured to initiate one or more transactions (Fig. 9, 10, "read to share"), wherein each of the one or more transactions comprises requests to access one or more data sources (Fig. 1, items 22A-D, memories in nodes 12A-12D); and one or more transaction managers configured to manage the one or more transactions initiated by the one or more applications (Col. 8, Lines 20-24, "memory operation...causing transfer of data from a source to a destination...within the initiator"; Col. 9, Lines 10-11, "Memory 22 is configured to store data and instruction code for use by processors"; Fig. 15A, "directory cache management unit"); wherein for each transaction (Fig. 9, 10, "read", "share", "own"), the one or more transaction managers are configured to request a read lock on a transaction freeze object (Col. 4, Line 26, "spin-lock operations" or Line 33-36, "barrier synchronization...waiting CPUs") to change the state of the transaction (Fig. 14, 15, "transaction blocking unit" or Fig. 7, item 166, "IDLE" or Fig. 8, item 188), and wherein the one or more transaction managers are configured to not change the state of the transaction without said lock (Col. 4, Line 26, "spin-lock operations"; Col. 4, Lines 42-44, "delaying the access of the last CPU").

However, Hagersten strongly suggests but does not expressly teach a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a

resume request by granting said read lock for said stored transaction freeze object (Fig.14, 15A, "transaction blocking unit", Col. 3, Lines 10-15, Lines 32-38, "spin lock").

Fowler as applied above teaches a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction; wherein a transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said read lock for said stored transaction freeze object (Col. 5, Lines 40-53,"pause/resume synchronization of the multiprocessor system"; Col. 8, Lines 52-56, "control signal to...effect resumption of program execution").

Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Oeltjen in view of Hagersten.

As to claim 26, Oeltjen merely suggests but does not expressly teach the method as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order ([35]).

However, Hagersten teaches the method as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Oeltjen and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Oeltjen and Hagersten because it provides for a

multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Oeltjen and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as suggested in Hagersten, Col. 4, Lines 45-49.

Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Oeltjen in view of Armangau et al, (US 6,659,992), hereinafter Armangau.

As to claim 28, Oeltjen does not expressly teach the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request if the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request.

However, Armangau teaches the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request (conditional) if the transaction manager is not paused and there are (interpreted to be a negative limitation) no outstanding state transition permission requests received prior to the pause request (Abstract, "permit the data to be removed...buffer becomes empty, the data mover retrieves the overflow").

Oeltjen and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Oeltjen and Armangau because it provides for

removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Oeltjen and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

Claims 35, 44 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Hagersten.

As to claim 35, Ault merely suggests service queried lock requests in FIFO order [35] but stops short of clearly teaching the limitation.

Oeltjen and Oliver do not expressly teach the method as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

However, Hagersten teaches the method as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Ault in view of Oliver and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as suggested in Hagersten, Col. 4, Lines 45-49.

As to claim 44, Ault merely suggests but does not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order ([35]).

Ault and Oliver do not expressly teach the computer readable storage medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

Hagersten as applied above teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

As to claim 53, Ault merely suggests but does not expressly teach the computer readable storage medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order ([35]).

Ault and Oliver do not expressly teach the computer readable storage medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

However, Hagersten as applied above teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Claims 30-32, 34, 36, 48-50, 52, 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault et al, (US 6,237,019), hereinafter Ault in view of Oliver, (US 6,029,190), hereinafter Oliver.

As to claim 30, Ault teaches a method, comprising: using one or more computers to perform; receiving a pause request (Fig. 3, Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue...by creating a queues"); pausing a transaction manager in response to the pause request by withholding read locks (supra) on a transaction freeze object (Abstract, Col. 4, Lines 20-25; Col. 9, Lines 46-57); receiving a resume request (Fig. 2, "ACCESS RESOURCE SEMOP(SEMID,+1); Fig. 3, "GRANT SEMAPHORE", item 316 meets a read lock); and resuming the transaction manager in response the resume request by granting locks on the transaction freeze object (Abstract, Col. 4, Lines 20-25; Col. 9, Lines 46-57); (Fig. 2, SEMOP(SEMID, -1)->"GRANT ACCESS", item 222).

However, Ault suggests but does not expressly teach a read lock in (Fig. 2, "SEMOP, SEMID, -1)->GRANT ACCESS") but does not clearly teach a "READ LOCK" necessarily occurs when accessing the semaphore (as READ can be reasonably interpreted as a verb and an adjective when any consideration of the semaphore variable); withholding read locks on a transaction freeze object that identifies a respective atomic transaction (Abstract, Col. 4, Lines

20-25; Col. 9, Lines 46-57); granting locks on the transaction freeze object that identifies a respective atomic transaction (Abstract, Col. 4, Lines 20-25; Col. 9, Lines 46-57).

Oliver teaches read lock and write lock management system based upon semaphore availability (see Title); withholding read locks on a transaction freeze object that identifies a respective atomic transaction (Fig. 1, item 112, see "READ LOCK...IS MUTEX AVAILABLE", where a MUTEX is necessarily and always an atomic transaction); granting locks on the transaction freeze object that identifies a respective atomic transaction (Fig. 2, "WRITE LOCK....ARE MUTEX AND SEMAPHORE SIMULTANEOUSLY AVAILABLE").

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for "read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location" as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for "read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location" as suggested in Oliver, Abstract.

As to claim 31, Ault teaches the method as recited, wherein a transaction freeze manager grants and withholds the read locks (Fig. 3, item 360, "WAIT QUEUE FOR INTERNAL LOCK").

However, Ault suggests read locks (Fig. 2) but does not clearly teach "READ LOCK".

Oliver as applied above teaches read lock (see title).

As to claim 32, Ault teaches the method as recited; wherein the transaction freeze manager is a part of the transaction manager (Fig 2, 5A-C).

As to claim 34, Ault teaches the method as recited, wherein the transaction freeze manager is configured to queue received lock requests in the order received (Fig. 3, "wait queue", 5A).

As to claim 36, Ault teaches the method as recited, wherein the transaction freeze manager is configured to grant a read lock if the transaction manager is not paused (Fig. 3, 5A).

However, Ault does not expressly teach "read lock".

Oliver as applied above teaches read lock (see title).

As to claim 48, Ault teaches a computer readable storage medium storing program instructions (Fig. 3, 5A-5E), wherein the program instructions are computer-executable to: receive a pause request (Fig. 2, "suspend"); pause a transaction manager in response to the pause request by withholding locks (Fig. 5a, "do forever" necessarily withholds locks) on a transaction freeze object receive a resume request (Fig. 2, "resume waiters"); and resume the transaction manager in response to the resume request by granting locks on the transaction freeze object (Fig. 2, "semop(semid, -1)->grant access").

However, Ault suggests read locks (Fig. 2) does not clearly teach read locks.

Oliver as applied above teaches read locks (see title); withholding read locks on a stored transaction freeze object that identifies a respective atomic transaction (Fig. 1, item 112, see "READ LOCK...IS MUTEX AVAILABLE", where a MUTEX is necessarily and always an

atomic transaction); granting read locks on the stored transaction freeze object that identifies the respective atomic transaction (Fig. 2, "WRITE LOCK....ARE MUTEX AND SEMAPHORE SIMULTANEOUSLY AVAILABLE").

As to claim 49, Ault teaches the computer readable storage medium as recited, wherein a transaction freeze manager grants and withholds the locks (Fig 2, 5A-C).

However, Ault does not clearly teach read locks.

Oliver as applied above teaches read locks (set title).

As to claim 50, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is a part of the transaction manager (Fig 2, 5A-C).

As to claim 52, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to queue received lock requests in the order received (Col. 2, Lines 40-43, "in order for the kernel semaphore logic to atomically update the semaphore value and maintain the wait queue").

As to claim 54, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 3, 5A).

However, Ault does not clearly teach read locks.

Oliver as applied above teaches read locks (set title).

Claims 33, 38, 42, 51 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Hagersten.

As to claim 33, Ault teaches the method as recited, wherein the transaction freeze manager is configured to receive requests (intended use) for locks on the stored transaction (Fig. 2, 5A-5C) freeze object to pause the transaction manager from an administrative entity.

However, Ault does not expressly teach write locks.

Oliver as applied above teaches write locks (see Title).

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as suggested in Oliver, Abstract.

Also Ault and Oliver do not expressly teach the transaction manager from an administrative entity.

However, Hagersten teaches an administrative entity (Fig. 9-10, "ADM....administrative").

Ault in view of Oliver and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and

Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transaction as suggested in Hagersten, Col. 4, Lines 45-49.

As to claim 38, Ault does not expressly teach the method as recited, wherein the transaction freeze manager is configured to not grant locks if (optional conditional) a lock on the transaction freeze object is currently held by an administrative entity.

However, Ault merely suggests a write lock (Fig. 2) but does not expressly teach a write lock.

Oliver teaches a write lock (see Title).

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as suggested in Oliver, Abstract.

However, Ault and Oliver do not expressly teach wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze object is currently held by an administrative entity.

Hagersten as applied above teaches wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze object is currently held by an administrative entity (Fig. 9, 10"ADM...Administrative").

As to claim 42, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager (Fig. 2, 5A-5C).

However, Ault and Oliver do not expressly teach pause from an administrative entity.

However, Hasgersten teaches pause from an administrative entity (Fig. 9).

As to claim 51, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to receive requests for (intended use) locks on the transaction freeze object to pause the transaction manager (Fig. 2, 5A-5C).

Ault suggests but does not expressly teach write locks (Fig 5A-5C).

However, Oliver as applied above teaches write locks (see title).

Ault and Oliver do not clearly teach an administrative entity.

However, Hagersten as applied above teaches an administrative entity (Fig. 9-10, "ADM....administrative").

As to claim 56, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze

object is currently held by an entity (Fig 2, 5A-5E); pause the transaction manager from an entity (Fig. 2, 3).

However, Ault does not expressly teach write lock.

Oliver as applied above teaches write lock (see title).

However, Ault and Oliver do not expressly teach administrative entity.

Hagersten as applied above teaches administrative entity (Fig. 9, "ADM...administrative").

Claims 57-58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hagersten in view of Fowler and in further view of Oeltjen et al, (US 2004/00225972 A1), hereinafter Oeltjen.

As to claim 57, Hagersten does not expressly teach and Fowler strongly suggest the system, wherein the transaction manager is configured to, without said permission, perform one or more operations associated with a current state of the given atomic transaction (Fowler, Col. 2, Lines 12-25, "means to display and modify memory").

However, Oeltjen teaches the system, wherein the transaction manager is configured to, without said permission, perform one or more operations associated with a current state of the given atomic transaction ([38], see "debugger....next, run, continue").

Hagersten in view of Fowler and Oeltjen are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Hagersten in view of Fowler and Oeltjen because it provides for a flow manager capability to save the sate of a flow and/or

test during execution and then restore the state of flow upon continued execution as discussed in Oeltjen, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Hagersten in view of Fowler and Oeltjen because it provides for a flow manager capability to save the sate of a flow and/or test during execution and then restore the state of flow upon continued execution as suggested in Oeltjen, Abstract.

As to claim 58, Hagersten does not expressly teach and Fowler strongly suggest the system, wherein the transaction manager is configured to, without said read lock for said stored transaction freeze object, perform one or more operations associated with a current state of the given atomic transaction (Fowler, Col. 2, Lines 12-25, "means to display and modify memory").

However Oeltjen as applied above teaches the system, wherein the transaction manager is configured to, without said permission, perform one or more operations associated with a current state of the given atomic transaction ([38], see "debugger....next, run, continue...other user full control over the location within the flow to manually set the current flow step and/or to establish flow nesting").

Claims 37, 46 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Armangau, (US 6,549,941), hereinafter Armangau.

As to claim 37, Ault teach the method as recited, wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 2, 5A-5E).

Ault does not teach the write and read lock.

However, Oliver teaches the write lock and read lock (see title).

Ault and Oliver do not expressly teach write lock and read lock and there are no outstanding lock requests prior to the lock request.

However, Armangau teaches there are no outstanding read lock requests received prior to the write lock request, and there are no outstanding read locks (Abstract).

Ault in view of Oliver and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for removing data from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

As to claim 46, Ault teaches the computer readable storage medium as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request if (optional condition) the transaction manager is not paused; wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 2-3, 5A-5E).

Ault does not expressly teach write lock and read lock.

However, Oliver teaches a write lock and read lock (see title).

Ault and Oliver do not expressly teach wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused; and there are no outstanding locks.

However, Armangau teaches there are no outstanding read lock requests received prior to the lock request, and there are no outstanding locks (Abstract).

Ault in view of Oliver and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

As to claim 55, Ault does not expressly teach the computer readable storage medium as recited, wherein the transaction freeze manager is configured to grant a write lock if the transaction manager is not paused (Fig. 2-3, 5A-5E)

Ault does not expressly teach write lock and read lock.

However, Oliver as applied above teaches write lock and read lock (see title).

Ault and Oliver do not expressly teach there are no outstanding read lock requests received prior to the lock request.

Armangau as applied above teaches there are no outstanding read lock requests received prior to the lock request and there are no outstanding read locks (see Abstract).

Conclusion

Applicant's amendment necessitated the amended citations (or new ground(s)) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

If applicant still believes there is patentable subject matter within the disclosure and has reasons why those differences define over the prior art, then applicant can look to MPEP § 324 IV (September 2007) and 37 CFR 1.114 for additional suggestions that may be helpful for overcoming the finality of this Office Action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph D. Wong whose telephone number is (571) 270-1015. The examiner can normally be reached on Monday through Friday, 10 AM – 6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain T. Alam can be reached on (571) 272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://portal.uspto.gov/external/portal/pair. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/JDW/

Asst. Examiner, Art Unit 2166

28 April 2009

/Khanh B. Pham/

Primary Examiner, Art Unit 2166